|  |  |
| --- | --- |
| Massively parallel flow cytometry data analysis pipeline using R Shiny  Employing bioinformatical approaches, machine learning and drawing. | |
|  | Organization  CIPHE-US012-INSERM  163, avenue de Luminy Case 936 13288,  Marseille, France  Direction  Hervé Luche, PhD  Scientific director - Immunophenotyping module at CIPHE  Abstract  Based on various R cytometry data analysis packages, many tools with clear UI were conceived in order to create a complete pipeline that allows a flexible and enriched analysis which respects the interoperability and the scalability of massively parallel flow cytometry single-cell data analysis for all the biologists without prior coding knowledge. |

|  |
| --- |
|  |

Table of Contents

[List of abbreviations: 2](#_Toc105194267)

[1. Introduction 4](#_Toc105194268)

[1.1. CIPHE 4](#_Toc105194269)

[1.2. Immunophenotyping Module: 4](#_Toc105194270)

[2. Materials and methods 4](#_Toc105194271)

[2.1. Data collection of a MPC assay 4](#_Toc105194272)

[2.2. Gating – CIPHE Gate 4](#_Toc105194273)

[2.2.1. Goal and preprocessing 4](#_Toc105194274)

[2.2.2. Gating menu 6](#_Toc105194275)

[2.2.3. Population annotation 6](#_Toc105194276)

[2.3. Equally sampled populations – CIPHE EqualSampling 6](#_Toc105194277)

[2.4. infinityFlow pipeline – CIPHE Infinity 6](#_Toc105194278)

[2.4.1. State of the art 6](#_Toc105194279)

[2.4.2. Data visualization 8](#_Toc105194280)

[2.4.3. Statistics tool 8](#_Toc105194281)

[2.5. Data exploitation 8](#_Toc105194282)

[3. Results 8](#_Toc105194283)

[3.1. Removing marginal and artefactual events 8](#_Toc105194284)

[3.2. Equally sampling 10](#_Toc105194285)

[3.3. Infinity output 10](#_Toc105194286)

[3.3.1. Parameters 10](#_Toc105194287)

[3.3.2. Visualization 10](#_Toc105194288)

[3.4. Statistical tests 10](#_Toc105194289)

[3.4.1. Non parametrical approach 10](#_Toc105194290)

[3.4.2. PCA and OPLS-DA 10](#_Toc105194291)

[4. Discussion 12](#_Toc105194292)

[4.1. Designing a panel for infinity flow 12](#_Toc105194293)

[4.2. Communicating with the package creators 12](#_Toc105194294)

[5. Conclusion 12](#_Toc105194295)

[Annex 13](#_Toc105194296)

[Personal GitHub: 13](#_Toc105194297)

[Team presentation: 13](#_Toc105194298)

[Example of use cases: 13](#_Toc105194299)

[References 14](#_Toc105194300)

# List of abbreviations:

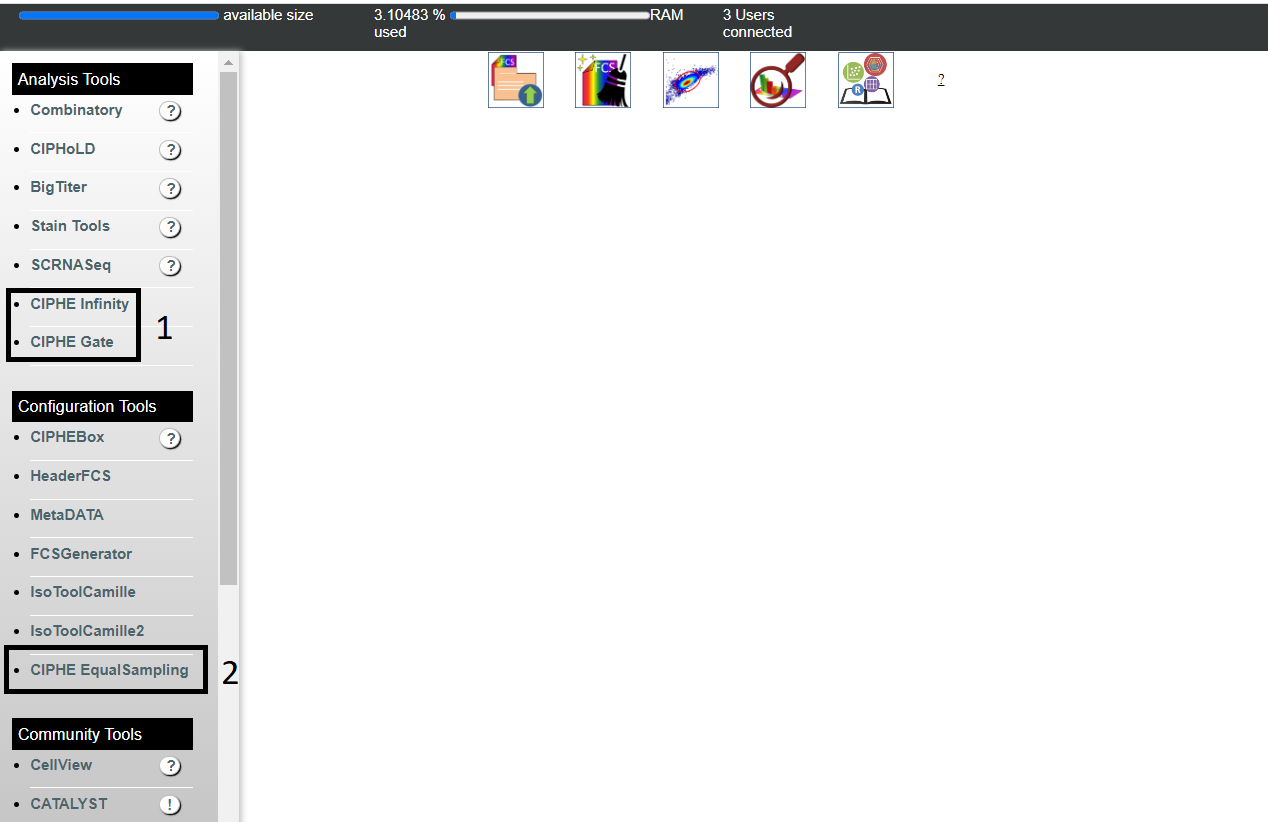
|  |  |
| --- | --- |
| Abbreviation | Explanation |
| AMU | Aix-Marseille University |
| CIPHE | Center of Immunophenomics |
| CNRS | National Centre for Scientific Research |
| CSV | Comma separated values |
| FCS | Flow Cytometry Standard |
| Inserm | National Institute of Health and Medical Research |
| MFI | Mean Fluorescent Intensity |
| OPLS-DA | Orthogonal Projections to Latent Structures Discriminant Analysis |
| PCA | Principal Component Analysis |
| PDF | Portable Document Format |
| UMAP | Uniform Manifold Approximation and Projection |
| VIP | Variable Importance in Projection |

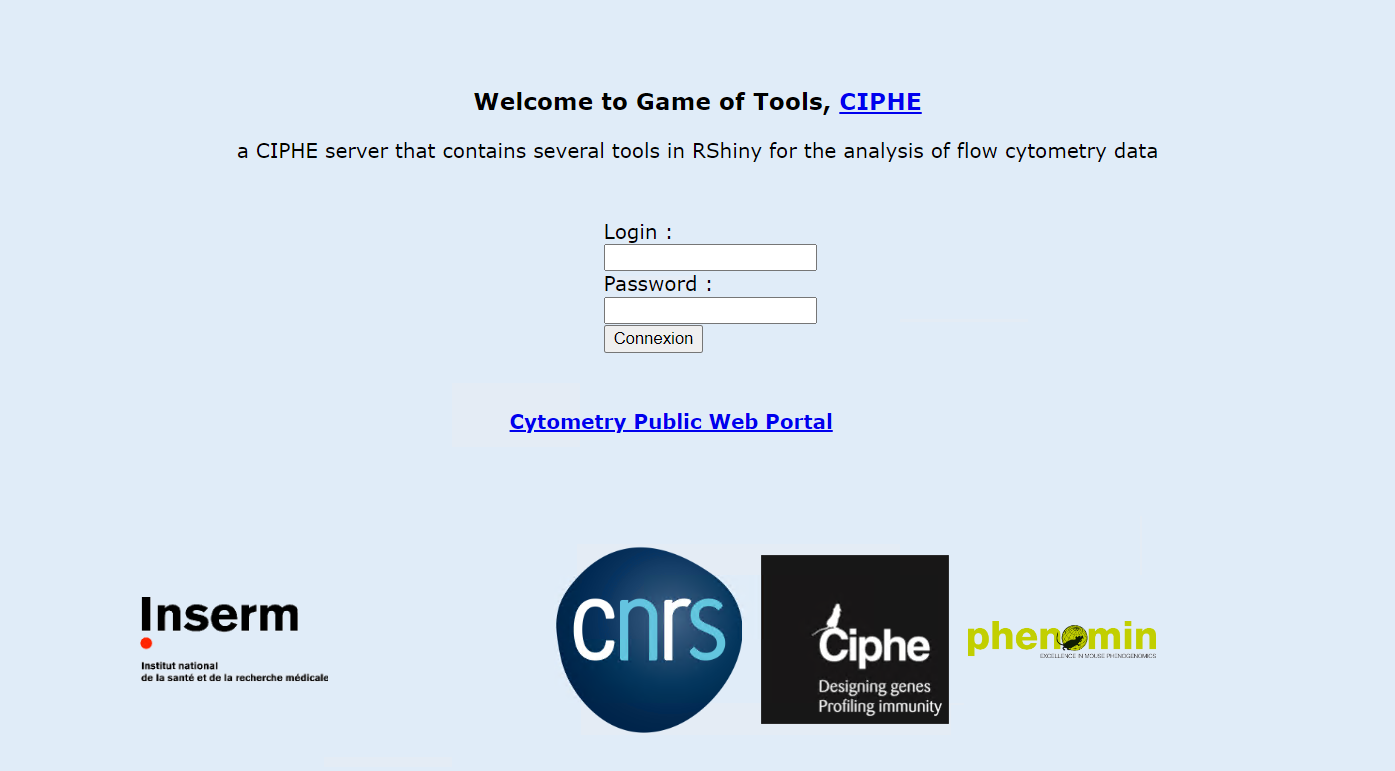
Definition of terms:

|  |  |
| --- | --- |
| Term | Definition |
| Compensation | Refers to the process of correcting a fluorescence spillover and removing the signal of any fluorochrome given to all detectors other than the one given to measure that dye. |
| Gating | Selecting an area on the scatter plot generated during the flow experiment that labels the cells as a group of population, each dot designs a single-cell. |
| GatingSet | Data structure that holds many type of gates on a collection of FCS files. |
| Invert gating | Also known as “NOT Boolean gating”; it creates a population that includes all cells that are outside a selected gate |
| Logicle (transformation) | A type of biexponential hyperbolic sine transformation that is used for cytometry data.providing many advantages over linear and log transformations. |
| Transformation | Cytometry data has been transformed using many types of transformation (arcsinh, logicle…) to improve visualisation and separation of negative and positive events into discrete populations. |

|  |  |
| --- | --- |
| Hervé Luche | Scientific director |
| Lilia Hadjem | Research engineer |
| Magali Grange | Research engineer |
| Marielle Mello | Research engineer |
| Priscilla Canavese | Research engineer |

**Figure 1 – R&D Team at CIPHE**

Presentation of Research and Development team members and their roles, all of the members are specialized in cytometry studies.

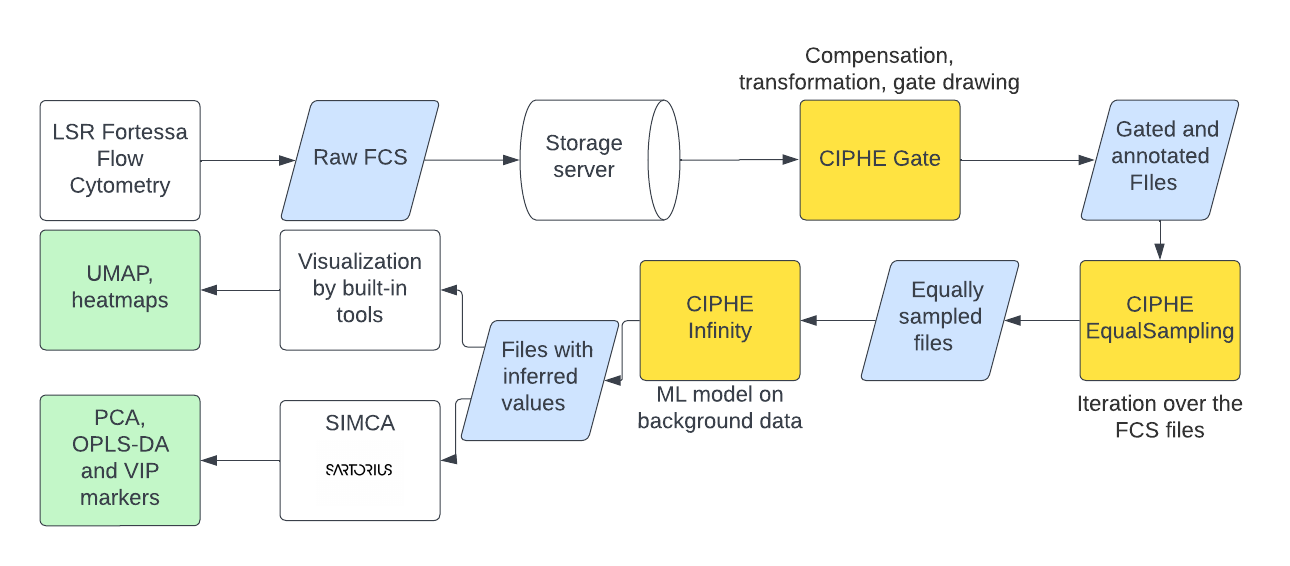


**Figure 2 – CIPHE server interface**

A

A

(LEFT) Web-based client to access the tools, available only for the PCs connected to the CIPHE network.

****(RIGHT) Overview of all the available tools, added analysis tools for the new approach: (1) CIPHE Infinity – CIPHE Gate - (2) CIPHE EqualSampling mini-tool.

**Figure 3 – Procedure pipeline**

Graphical presentation of the different sequences of the pipeline conceived with the moment of employment of each of the tools. By using indexed and equally sampled files for the infinity flow analysis, the approach is supervised and will equally present the rare populations into the ML model.

# Introduction

## CIPHE

**CIPHE** is a mixed unit that opened on the Luminy site in Marseille in 2012. It’s a unit of Inserm (US012), CNRS (UMS3367) and AMU, which focuses on the interactions and responses of the immune system under different conditions: basal, inflammatory, or in the context of a pathology, with or without administration of treatment. CIPHE, being a service platform for private and academic laboratories, conducts research and development projects in order to offer the best possible services allowing the reproducibility of the experiments and limiting as well the variability of the inter-operators results.

## Immunophenotyping Module:

The objective of this module is to analyze the phenotype of the immune system by targeting immune populations using antibodies (Ab) that recognize specific antigens expressed on target cells. These analyses are performed using flow and mass cytometry techniques. The immunophenomics team at CIPHE consists of four research engineers specialized in cytometry analysis under the direction of Dr. Hervé Luche (**Figure 1**). The lab has two local servers (dev & prod) for hosting tools based on R Shiny (**Figure 2**), also it has many servers dedicated for data storage.

**Massively parallel flow cytometry** assays yield a huge amount of data with around 64 million unique events (A.K.A single cell measurements). Infinity Flow [(Becht *et al.*, 2020)](#infinityflow) allows the analysis of hundreds of cell-surface proteins using machine learning models mainly based on XGBoost [(Chen and](#XGBOOST) Guestrin, 2016) implementation of gradient boosted trees to impute values of expression due to its accuracy at both high and low cell numbers and speed. The goal is to build an easy-to-use web interface that allows the biologists with no prior coding knowledge to take advantage of this tool. However, it is necessary to manipulate (transform, compensate…) the raw data, clean up and exclude all the outlier events that can ruin the experiment with bad data, index the cells in case of a supervised approach. In the CIPHE lab, the team already had an artefactual R Shiny gating tool, my decision was to make a new one that takes advantage of a new R package [(R Core Team, 2021)](#R) called “cytoExploreR” [(Hammill, 2021)](#cytoexploreR) which made gating easier. Furthermore, by equally sampling the annotated files, rare populations will be equally presented to the machine learning model built by Infinity Flow, thus the algorithm will identify them well. By conceiving such a pipeline (**Figure 3**), different possible combinations of the approach could be made. There is a lot of room for flexibility in the usage of the tools

# Materials and methods

## Data collection of a MPC assay

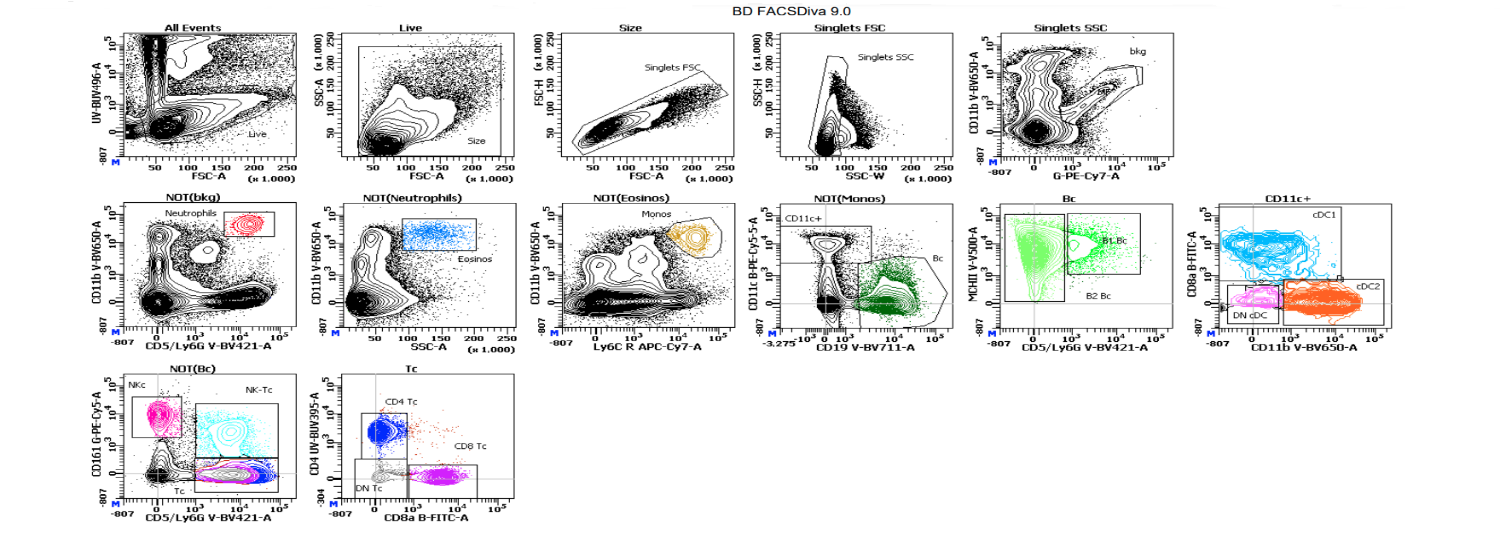
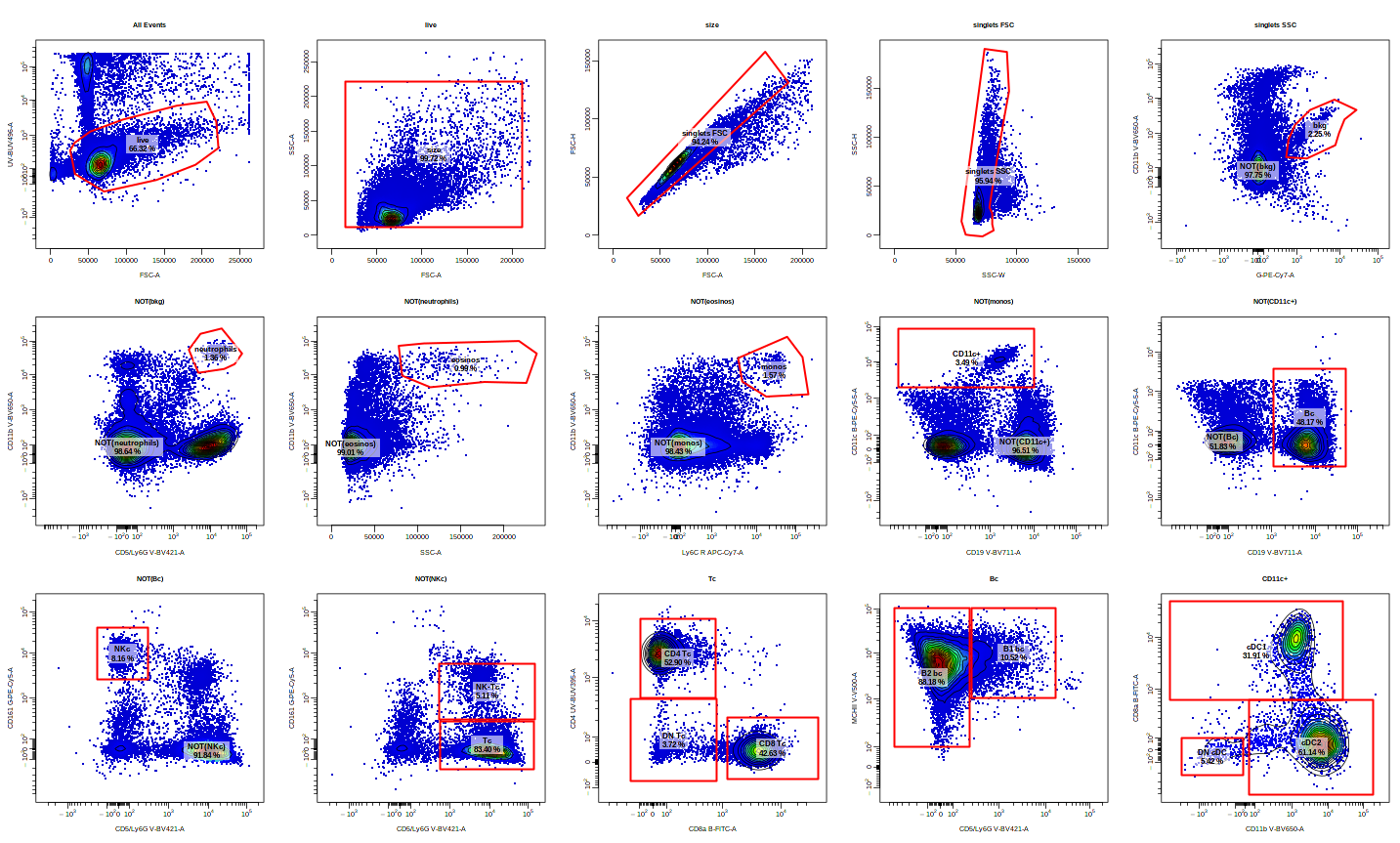
The experiment used in the analysis was done in CIPHE Lab in 2018, using LEGENDScreen Mouse PE Kit (cat. 700002; Biolegend, San Diego, California, United States) in 3 plates of 96 wells for the purpose of phenotyping and capturing the cell surface markers by using the flow cytometer LSR Fortessa. The dataset is made of 269 files in .fcs format and each containing in average 200 000 events of raw measurements. The files are separated in subfolders for each plate and each file is annotated with the well it belongs to and all the events measured in these folders are raw and require some type of filtering that cleans the data and eliminates events that shouldn’t be present in the downstream analysis.

The analysis was then done manually in order to infer which marker is expressed by which population, that is time consuming and prone to variations as gates are subjectively set on every single file by the operator analyzing the data.

## Gating – CIPHE Gate

### Goal and preprocessing

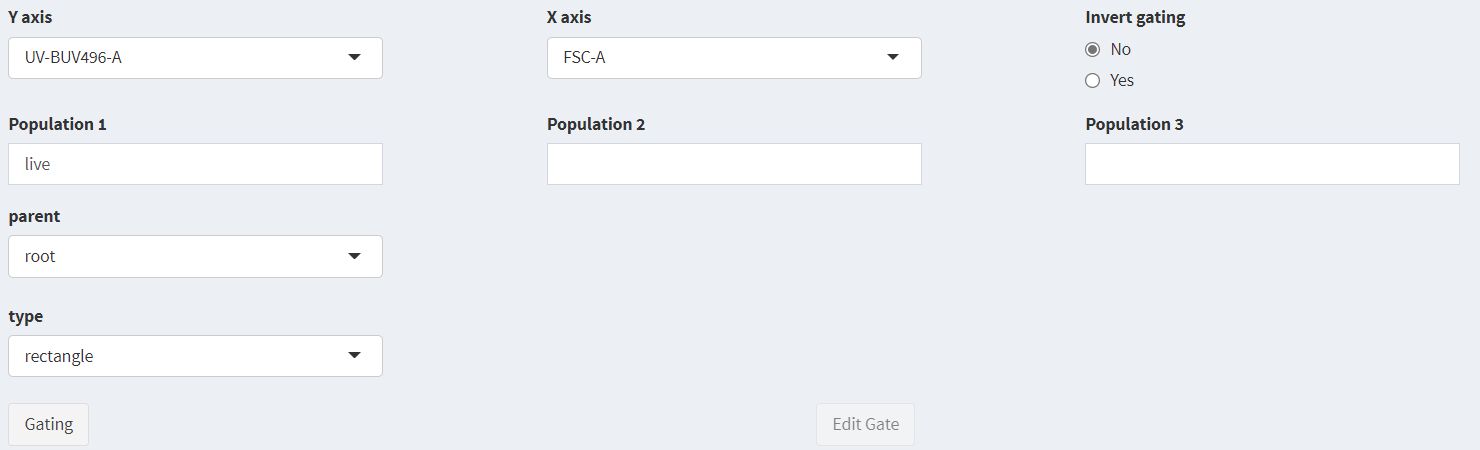
For the purpose of filtering the unwanted events, a step of gating was introduced on raw data. I worked on creating a tool that is based on the R package CytoExploreR that has the purpose of constructing gates on the data in an interactive manner based on R and R shiny as the web presentation of the tool with different shapes of gating. The result was an interactive tool that can create a gating template from scratch, or accept two types of gating templates either generated by the tool itself (.CSV) or generated by a professional cytometry flow data software like FlowJo.( .WSPT). If a gating template was provided



**Figure 4 – BD FACSDiva VS CIPHE Gate**

a

B

Re-creation of the gating analysis done on (A) BD FACSDiva 9.0 over in (B) the CIPHE Gate tool in order to apply it to all the raw FCS files of the experiment.

**Figure 5 – CIPHE Gate gating menu**

Presentation of the menu of CIPHE Gate:

* Y and X axis depending on the name of columns
* Invert gating radio button (To plot NOT(….) gate)
* User Population names up to three
* Parent gate to plot the selected population on
* Type of gates: (Rectangle, Polygon, Ellipse…)
* Plot a new gate or edit an old one

to the tool, this gating template will be directly applied to all the FCS files uploaded into the tool (**Figure 4**). The tool also allows the **preprocessing** step of the data including compensation and logicle transformation [(Moore and Parks, 2012)](#logicletrans). It’s possible to view the setup of the experiment inside the tool, moreover a mix of a combination of the FCS files is presented to show the application of the gating template over all the data thus the drawing of the gates will not be biased by one of the wells or plates that may arise during acquisition.

### Gating menu

The Shiny menu (**Figure 5**) allows for selection of the options of the gates in an interactive manner. The buttons of editing and plotting gates are enabled and disabled depending on the choices of the user. Also, an additional tab was conceived to view each of the files individually, in case the user wants to check for outlier files in the dataset uploaded after the application of the gating template. Since checking for each of the drawings individually is time consuming, an output of excel file containing the statistics for all of the gating sets can be downloaded. Upon statistical analysis, the user can detect for any outliers with different statistics (count, MFI, percentage…) for each gate in the GatingSet indicating the abundance of the populations in each of the gates drawn. In addition, the user has the choice to export the gating template in FlowJo (v.10.8) [(Becton, Dickinson and Company, 2021)](#becktonflowjo) (powered by an instance of FlowJo in docker of the Ubuntu local server at CIPHE) or cytoML format [(Finak, Jiang and Gottardo, 2018)](#cytoML).

### Population annotation

An extra layer was added to this program, in order to index the events of populations observed in the samples. In hindsight, this function is a simple concatenation of all the events of a user-specified gate with an added column indicating the Annotation ID of the chosen populations (). A limitation of the FCS file format is that the columns can only contain numbers. Hence, the index has to be a number and not a string (text). As output, a CSV file indicates what population belongs to which ID. The modified version of the R package that powers this tool “cipheCytoExploreR” can be found on GitHub, it contains some changes to allow for the compatibility of the methods with R Shiny web UI.

## Equally sampled populations – CIPHE EqualSampling

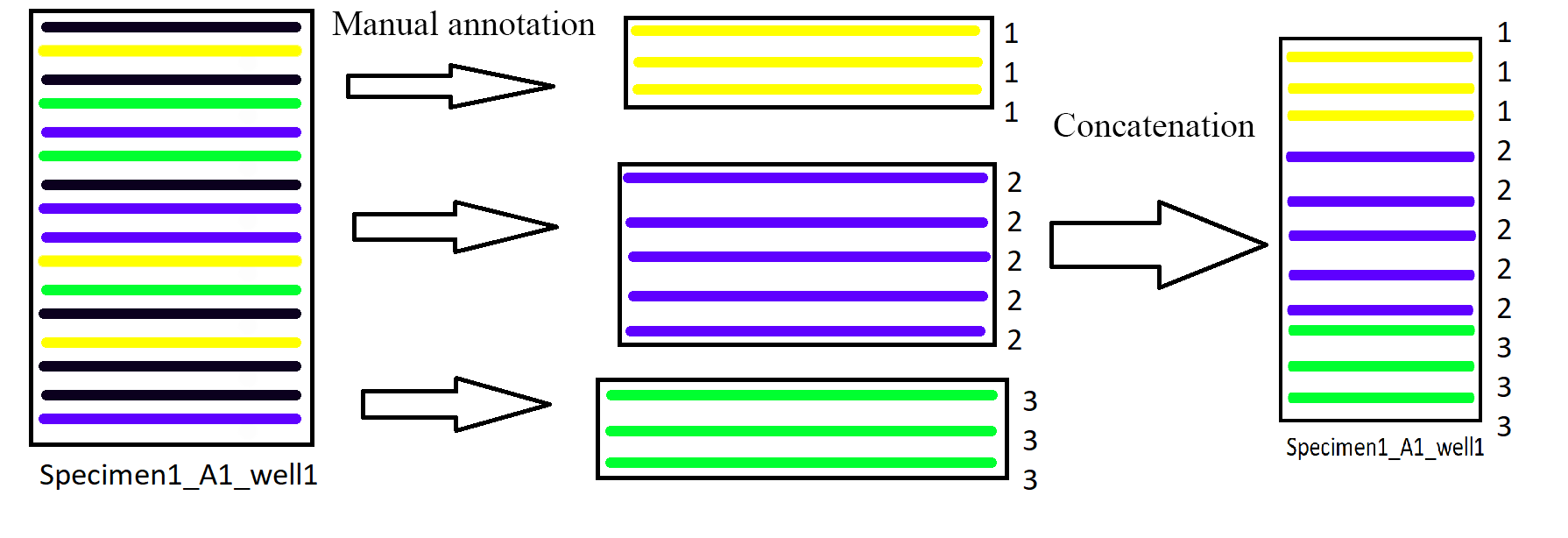
Immune cells are heterogeneous and present in widely different ratios inside the samples. In order to present the different type of cells in an equal way to the machine learning algorithm so it could infer the right values for the predicted data, the number of cells in each population should be equal. In this approach, the value to be based on will be the number of the least presented population in each file. Therefore, based on an iterating function over each file, all the populations will be sampled based on the lowest number of presentation for the purpose of having an equally presented mix of populations from each file. In order to make the tool flexible, the user should select the column that he would like to equally sample his files

Alternatively, the ratio of all the different populations can be kept as they are, but then the weights of the machine learning algorithm should be modified to give more weights of learning for the least present population. Thus, inferred data will not be biased by over-represented populations.

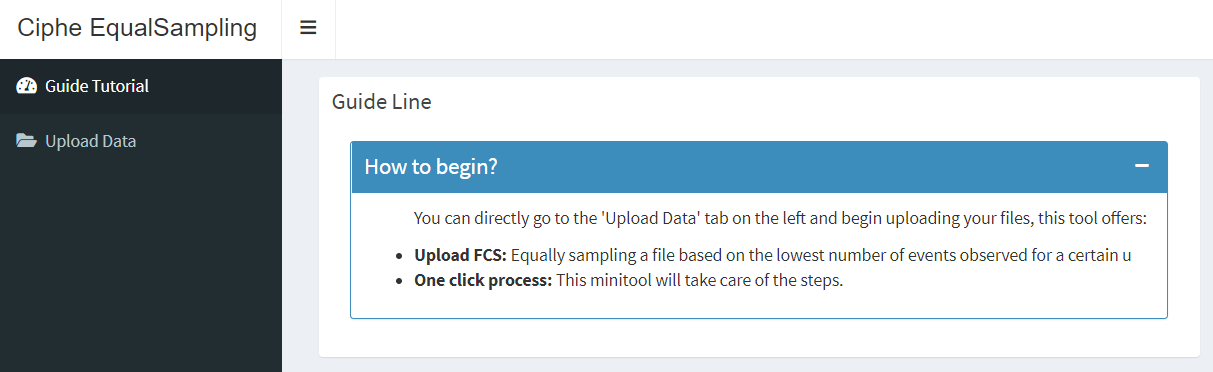
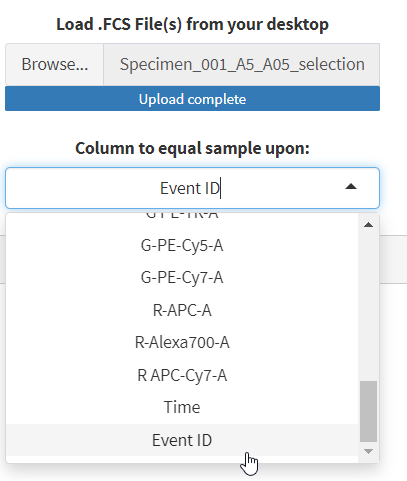
## infinityFlow pipeline – CIPHE Infinity

### State of the art

Based on the infinityFlow R package, the main idea is to infer the value of each exploratory marker for the studied immune cell samples. The inferred value will be based on the specific exploratory markers assigned for each well and its raw intensity measurement. Then, a XGBoost based machine learning algorithm will split the data into a training set (50% of the events) and a test set to evaluate performance (50% of the events). The package allows for a robust number of events sampling for each input files. This was modified in CIPHE Infinity to be expressed in % of the average size of events: rather than a robust number because not all the FCS files contain the same number of events. To make up for the background interference with the cytometry laser measurements, the pipeline will then create a background corrected file which practically assigns all the values of the measured isotypes to 1 and then scales the corresponding markers inferred values to register a more accurate value of

****

**Figure 6 - Illustration of the procedure of annotation**

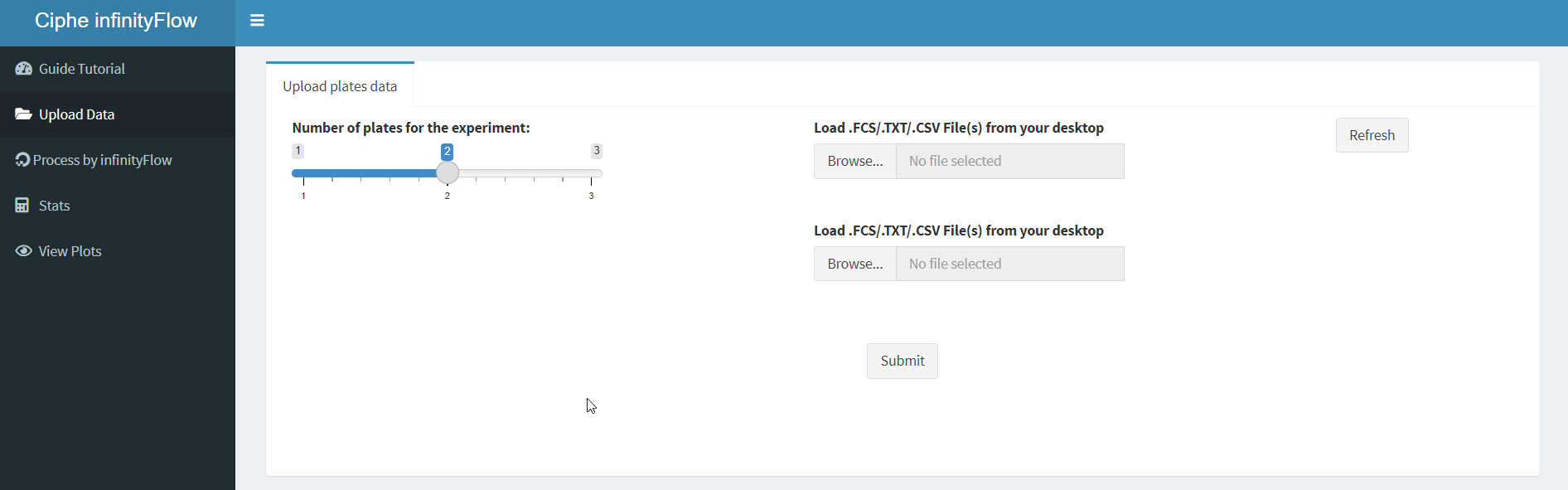
****The logic of indexing the files: based on the gating template in use, the user will select which populations he would like to gate for his output. The selected populations will be extracted from the gates with an extra column indicating the index of the population it belongs to. When this is done to all the populations, the matrices will be concatenated into one.

**Figure 7 – CIPHE EqualSampling UI**

Screenshots from CIPHE EqualSampling tool:

(LEFT) Main menu: small tutorial for the users.

(RIGHT) Column selection allowing for flexibility in the EqualSampling factor-column, in this case the column indexing for population in the FCS file is called “Event ID”.



**Figure 8 – CIPHE Infinity “Upload data” tab**

Based on the number of plates of the experiments, the user gets the choices to upload his data. Additionally, on the left there are other tabs that allow for a lot of functionalities that can be used on directly on the infinityFlow output or on an archived result of the pipeline.

expression that ignores best the interference of the background measurements with the real exploratory marker measurement. The purpose was to create a R Shiny tool to run these of analyzes (**Figure 8**).

### Data visualization

In addition to that, the infinityFlow pipeline runs a UMAP [(Becht et al., 2018)](#UMAP) dimension reduction step on the background measurements that groups together the cells that have similar background measurements. This way the original input fcs files will be further enriched with columns presenting the XGBoost inferred values and UMAP1 and UMAP2 coordinates. In the tool conceived, it was decided to visualize the data in two different ways (Heatmap and UMAP). The heatmap is based on the R package ComplexHeatmap [(Gu, Eils and Schlesner, 2016)](#complexheatmap) that allows for euclidian hierarchical clustering on both of the markers selected by the user and the annotated populations (by CIPHE Gate). The choices of the markers and populations are done by the user, the tool allows for the selection of multiple columns (in this case, each FCS file had 292 columns).

On the other hand, for the UMAP, ggplot2 [(Wickham, 2016)](#ggplot2) package was used for interactively plot two graphs of the UMAP each time, the first one on top representing the UMAP colored by the ID of the annotation conceived in a flexible way to accept different types of clustering (supervised or not) and another one colored by a user-selected column values, the values are then graded from blue to red, the values belonging to the upper 0.85 quantile will be colored in red, indicating a well expressed marker.

### Statistics tool

Finally, population-wise statistics can be generated using this tool, many events from all the presented plates will be sampled in order to calculate different parameters of descriptive statistics for each population including: median, MFI, average, mode…

## Data exploitation

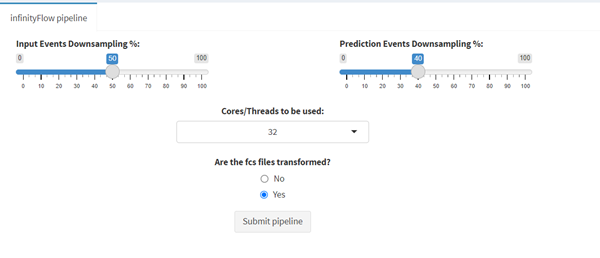
In order to make sense of the data generated and find specific patterns and signatures for the studied populations, the last step of this approach consisted of using different statistical methods in order to infer some meaning for the data. The average expression value for each pop on 10 randomly selected output files were studied and ran through a PCA (Worley and Powers, 2016) test and OPLS-DA [(Bylesjö *et al.*, 2006)](#OPLSDA) in the purpose of finding the most meaningful variances in the exploratory markers that explain the differences between the populations. The reason to consider the average of the population’s events instead of each cell on its own is to lessen the intra-population variations and the bias that can be produced by a wrong measurement or an outlier. Only then the “VIP” [(Akarachantachote, Chadcham and Saithanu, 2014)](#vip) variables (markers) correlated to each annotation can be extracted.

The statistics are generated by R each row contains the UMAP coordinates along the Annotation ID (for population) and the file ID with all the other exploratory inferred values. Then the analysis is done using third party software like “SIMCA”[(*SIMCA® - Multivariate Data Analysis Software | Sartorius*, n.d. version 17.0.2)](#simca).

# Results

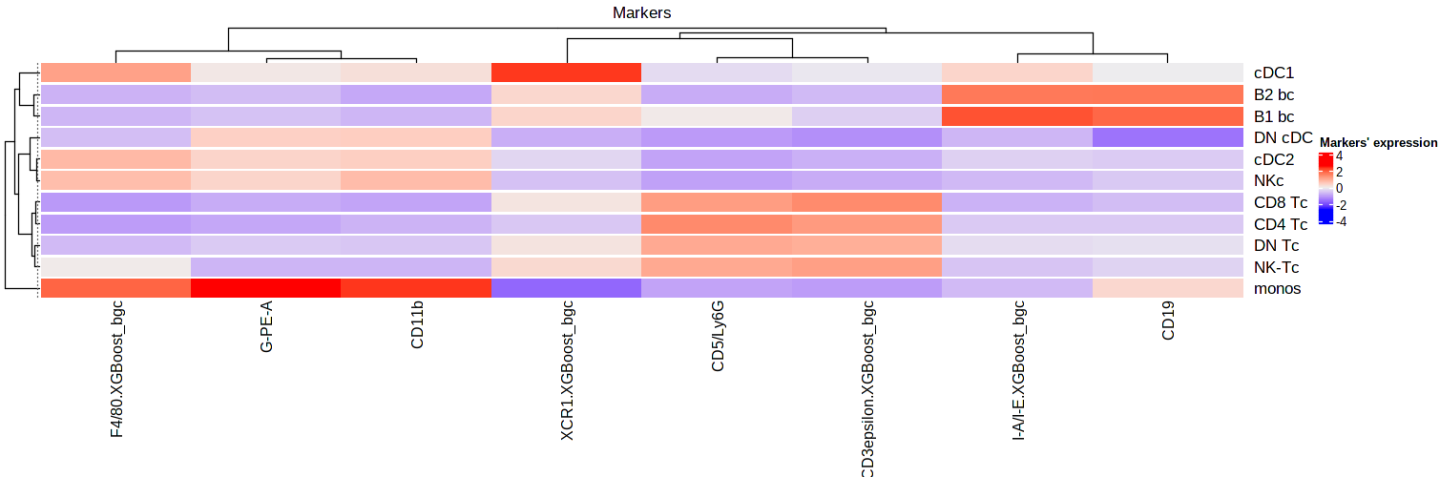
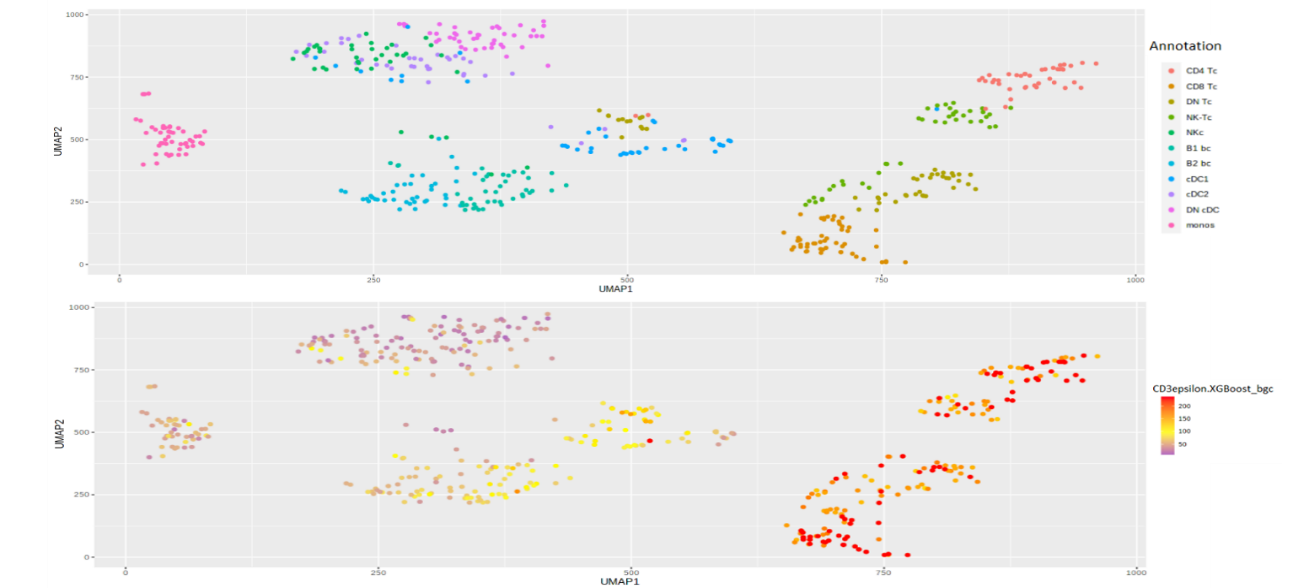
## Removing marginal and artefactual events

The raw output of the flow cytometry experiment consisted of 266 files averaging 200 000 events non-transformed and non-compensated. Because of the artifacts that can be generated in acquisition, the gating part of the pipeline eliminates dead and doublet cells thus reducing the percentage of cells into around %66 of all the events. The populations that make up the samples are then annotated by continuing the gating strategy as shown in **Figure 4**.

****

**Figure 9-InfinityFlow pipeline parameters UI**

Snapshot of the parameters used in the analysis of the real dataset. Setting the radio button to “Yes” will skip the logicle transformation and re-transformation of the data by infinityFlow which will save time (having the gated data already transformed).



**Figure 10 – Visualization tools offered by CIPHE Infinity**

Figures showing an example of two visualization methods generated by CIPHE Infinity tool (A) Two UMAP plots comparing between the populations and the expression of CD3epsilon marker on the cells. (B) A heatmap with all the populations as rows and columns a handful of “signature” markers showing good results, also the horizontal and vertical hierarchical clustering of the populations and markers respectively.

B

a

## Equally sampling

Based on the number of the least represented population in the sample (DN cDC in this case going from around 180 – 500 per well), the other populations will be sampled by the same number. As a result, the fcs files will contain each between 1980 and 4120 events, let the number of average events after the equally sampling step be around 3050 events all the way down from 200 000; equivalent to %1.5 of the initial raw data. Additionally, the data matrix is annotated by the index of population it belongs to in order to carry biological insight in downstream analysis of the pipeline.

## Infinity output

Evidently by reducing the number of events studied, the execution time of the infinityFlow method drastically decreases in both aspects of training the model and imputing the expression value of each of the markers.

### Parameters

In this run, %50 of the whole files were used as samples and %40 of the sample as buildups for the machine learning model (**Figure 9**). The model was then validated by plotting the UMAP coordinates that are based solely on the background channels and coloring the dots depending on the Annotation ID of each single cell observation. As mentioned in the “methods” part, there are two types of FCS output folders, one that is background corrected for the isotypes and the other one that isn’t. We focused on analyzing the results of the background corrected files because they have done the purpose of reducing the background noise to signal ratio.

### Visualization

The data can then be visualized by using the same tool (CIPHE Infinity). Choosing to plot a UMAP or a heatmap, both will serve for the purpose of visualizing the data distribution. By looking at some markers that are known that should be shown in some of group of populations of the cells (Ex: CD3epsilon marker on T cells) can indicate if the data inferring method have done well (**Figure 10**).

## Statistical tests

### Non parametrical approach

When looking at the descriptive statistical resulting excel file, it contains 258 columns each one of them representing one marker. In order to check the most important markers that make the variations between the populations, a non-parametrical **Kruskal–Wallis test** [(McKight and Najab, 2010)](#kruskalwallis)was initialized on all the columns with the null hypothesis being that all the samples belong to the same population, which was proven false for the vast majority of the columns ( p-value < 2 \* 10 ^-6).

This approach doesn’t allow for the comparison between the impact of all the markers on the partition between the populations, so a different method was needed, hence the use of “**SIMCA**” for **PCA and OPLS-DA.**

### PCA and OPLS-DA

The initial use insisted of randomly choosing 20 single-cell events for each population from all the files and see the extracted features considered between the expressions of the markers, but based on PCA and OPLS-DA. It was observed that the variations and noise between the measurements inside the same population will yield a lot of outliers for the variables studies (**Figure 11**), thus the same test was done on averages. The method begins by generating an average of expression for all the markers of all the populations in a certain number of FCS files (10 files in this example) chosen from all the plates to avoid plate-specific bias. By calculating the averages, the noise between the annotated events belonging to the same population is reduced. Therefore, the data is more reliable to fish out specific signals and signatures (**Figure 12**). In order to fish out the most important markers of the group of T cells, the next analysis will be focused only on the annotated T cells (Annotation ID 1, 2, 3, and 4) that make a sub-group inside the whole sample. The OPLS-DA plot will result in R(X)2 = 0.783, R(Y)2 = 0.924 and Q2 = 0.848 of this sub-group and then it’s possible fish out the VIPs correlated to the T cells annotations for the four populations as shown in with a robust threshold of a score of **1.2** (**Figure 13**). Based on the bibliographical knowledge the results indicate the success of this approach in filtering for columns that explain the differences between the populations.

****

**Figure 11-- PCA on random selection of cells**

The data presents some outlier dots. Also the correlation between the different populations is hard to understand directly as some populations show noisy distribution.

**Figure 12 – Visualizations done on the average of 10 FCS files**

PCA (LEFT) and OPLS -DA (RIGHT) done on the average of 10 FCS files, showing better score of correlation and a more clear distribution of populations in respect of the expression of their markers

**Figure 13 – Visualization and VIP on T cells**



Plot of OPLS-DA (LEFT) done only on the T cells (populations 1 to 4) further showing the differences and discrimination between these populations. (RIGHT) VIP Markers that make the biggest variances between the different quadrants of the circle representing the distribution of the populations.

# Discussion

The data generated by a MPC assay is enormous, but the data that is then generated by the infinityFlow method is even more complicated because for each well, there will be an exploratory antibody, which will then translate into an extra column inserted in all of the FCS files indicating the expression of this particular marker in all the events observed. Given the wealth of information, affordability, and compatibility with standard flow cytometers provided by the Infinity Flow pipeline, it is an interesting way to test for the presence of exploratory markers on one or many more population of cells. In this report, just for the validation of the pipeline, we only focused on T cells and the markers that differentiate between them. But the results we obtained are very interesting and meaningful for all the populations and their markers tested in this approach. By using this intelligent approach based on machine learning algorithms, phenotyping cells have become easier and quicker than before using standard flow cytometry approach and without the need for gigantic computational power. Although some limitations were found while developing the tools, we were able to overcome these obstacles:

### Designing a panel for infinity flow

First of all the LEGENDScreen plates used do not entail all the isotype control for all the infinity markers tested in the kit like the “Rat IgG2a lambda” isotype, we had to manually modify the name of the isotype to “blank” in order to launch the pipeline, it should keep noted that when designing a panel with infinity flow in mind, the control isotype of the marker should be present obligatorily.

### Communicating with the package creators

Some of the functionalities that we were trying to add to our tools have been limited by both of the packages used. For example infinity Flow pipeline will expect non-transformed input of FCS files. Since we are working on a complete pipeline, it’s more time and computationally efficient to transform the data only once (before the gating step), which made me push an [issue](https://github.com/ebecht/infinityFlow/issues/6) to the original creator (Mr. Etienne Becht), I applied his modifications (mentioned in his reply) to our own modified version of the package “[cipheinfinityFlow](https://github.com/GeorgeAlehandro/cipheinfinityFlow)” by adding an optional argument (transform = T/F) and accepting FCS files as input instead of a directory. We faced a similar issue with the cytoExploreR package while plotting the gates but when using the tool on server-side non-interactive R session mode it will fail to do so. Therefore, I wrote to the creator (Mr. Dillon Hammil) who [promised](https://github.com/DillonHammill/CytoExploreR/issues/143) improvements in a future release to enhance the package’s compatibility with R Shiny client. Nevertheless, no issues were found when working with the tool on a local machine except that cytoML format doesn’t accept the invert gating booleanfilter that is written when a NOT( ) gate is introduced in cytoExploreR or in FlowJo format, which is a limitation if we ever want to export a gating strategy that contains an invert gate in cytoML format (optional) because it will throw an error. Nevertheless, using shinydashboardPlus package in the UI, the conceived application tools are made to be interactive and will warn the user about any errors.

# Conclusion

The main goal of the internship in the beginning was just the conception of a user-friendly graphical interface for the infinityFlow R package. Evidently, we ended up doing more than that by creating three different tools that can each be used on their own or be related to each other to create a complete pipeline that satisfies the purpose of phenotyping the immune cells. To validate our pipeline, we also ran a full-scale test on a real dataset of 60 000 000 single cells that used the three tools that we created beside a third-party tool like “SIMCA” to run PCA, OPLS-DA visualization and to fish out the most important variables that make the difference between the populations. What characterizes the tools is that they can accept the outputs and the templates of other software and so highlighting the concept of interoperability in the development-side of software in exchanging, making use of information, experiment validation and pipeline reproducibility.

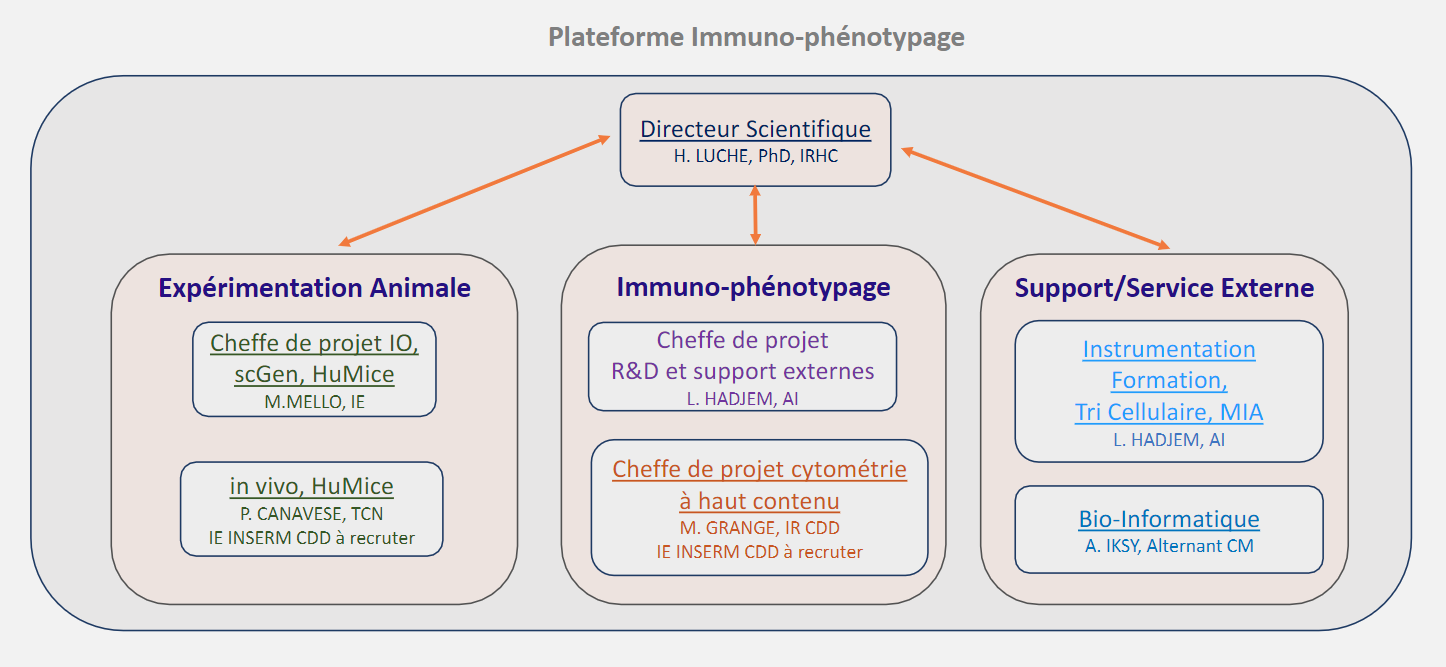
Conventionally, our tool “CIPHE Gate” highlights what is currently the state of the art in the area of gating using R, paired with “CIPHE EqualSampling” that will make sure that rare populations will be not overshadowed by the more abundant populations and finally all that goes into “CIPHE Infinity” which is an easy to use tool to launch cell-surface analyses that exploit the intelligence of machine learning in inferring the relations between the different populations based on flow cytometry data.

# Annex

## Personal GitHub:

Please refer to my [GitHub repository](https://github.com/GeorgeAlehandro/CIPHE_internship) for more information about the tools and their usage.

## Team presentation:



The figure above is a more detailed presentation of the Immunophenotyping module team at CIPHE, this is an official version but outdated.

For more details: click [here](https://github.com/GeorgeAlehandro/CIPHE_internship/blob/main/misc/Diagramme%20Fonctionnel-2022.pdf).

## Example of use cases:

The diagram above presents 3 different use cases to run CIPHE infinity analysis:

* Case 1: Used the 3 tools to organize, clean and annotate his FCS files (same as the paper’s example).
* Case2: The user had his files already annotated and decided to do EqualSampling before continuing.
* Case 3: Direct application of the infinity flow method on FCS files.

# References

Akarachantachote, N., Chadcham, S. and Saithanu, K. (2014) ‘Cutoff threshold of variable importance in projection for variable selection’, *International Journal of Pure and Apllied Mathematics*, 94. doi:[10.12732/ijpam.v94i3.2](http://www.ijpam.eu/contents/2014-94-3/2/).

Becht, E. *et al.* (2018) ‘Dimensionality reduction for visualizing single-cell data using UMAP’, *Nature Biotechnology* [Preprint]. doi:[10.1038/nbt.4314](https://www.nature.com/articles/nbt.4314).

Becht, E. *et al.* (2020) ‘Infinity Flow: High-throughput single-cell quantification of 100s of proteins using conventional flow cytometry and machine learning’. bioRxiv, p. 2020.06.17.152926. doi:[10.1101/2020.06.17.152926](https://www.biorxiv.org/content/10.1101/2020.06.17.152926v1).

Becton, Dickinson and Company (2021) [*FlowJoTM Software for Windows*.](https://www.flowjo.com/solutions/flowjo/downloads/tutorials)

Bylesjö, M. *et al.* (2006) ‘OPLS discriminant analysis: combining the strengths of PLS-DA and SIMCA classification’, *Journal of Chemometrics*, 20(8–10), pp. 341–351. doi:[10.1002/cem.1006](https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A152811&dswid=-7840).

Chen, T. and Guestrin, C. (2016) ‘XGBoost: A Scalable Tree Boosting System’, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. doi:[10.1145/2939672.2939785](https://arxiv.org/abs/1603.02754).

Finak, G., Jiang, W. and Gottardo, R. (2018) ‘CytoML for cross-platform cytometry data sharing’, *Cytometry Part A*, 93(12), pp. 1189–1196. doi:[10.1002/cyto.a.23663](https://pubmed.ncbi.nlm.nih.gov/30551257/).

Gu, Z., Eils, R. and Schlesner, M. (2016) ‘Complex heatmaps reveal patterns and correlations in multidimensional genomic data’, *Bioinformatics*, 32(18), pp. 2847–2849. doi:[10.1093/bioinformatics/btw313](https://pubmed.ncbi.nlm.nih.gov/27207943/).

Hammill, D. (2021) *CytoExploreR: Interactive Analysis of Cytometry Data*. Available at: <https://github.com/DillonHammill/CytoExploreR>.

McKight, P.E. and Najab, J. (2010) ‘Kruskal-Wallis Test’, in *The Corsini Encyclopedia of Psychology*. John Wiley & Sons, Ltd, pp. 1–1. doi:[10.1002/9780470479216.corpsy0491](https://www.researchgate.net/publication/278321713_Kruskal-Wallis_Test).

Moore, W.A. and Parks, D.R. (2012) ‘Update for the Logicle Data Scale Including Operational Code Implementations’, *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 81(4), pp. 273–277. [doi:10.1002/cyto.a.22030](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4761345/).

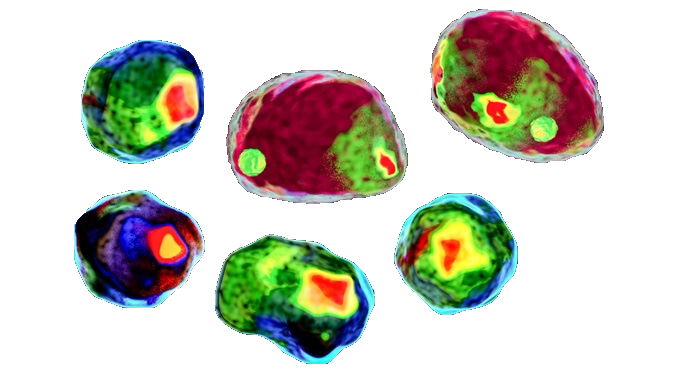
R Core Team (2021) *R: A Language and Environment for Statistical Computing*. Vienna, Austria (R Foundation for Statistical Computing). Available at: [https://www.R-project.org/.](https://www.R-project.org/)

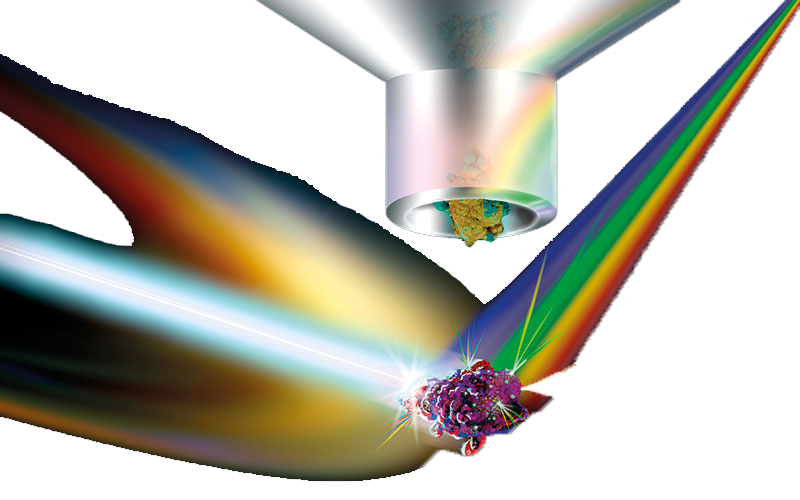
*SIMCA® - Multivariate Data Analysis Software | Sartorius* (no date). Available at: [https://www.sartorius.com/en/products/process-analytical-technology/data-analytics-software/mvda-software/simca](https://www.sartorius.com/en/products/process-analytical-technology/data-analytics-software/mvda-software/simca%20)

Wickham, H. (2016) [*ggplot2*](https://ggplot2-book.org/)*: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Worley, B. and Powers, R. (2016) ‘PCA as a practical indicator of OPLS-DA model reliability’, *Current Metabolomics*, 4(2), pp. 97–103. doi:[10.2174/2213235X04666160613122429](https://pubmed.ncbi.nlm.nih.gov/27547730/).

Summary

Using state of the art R packages for massively parallel flow cytometry data analysis, the goal of this report is to shed the light on a newly conceived pipeline that allows biologists with no prior coding knowledge to run their cytometry data cleaning, down sampling and cell surface analysis employing machine learning models. These tools also offer various visualization possibilities alongside statistics generation for data validation. The report also showcases a real exploitation of a dataset and the steps followed to extract biological meaning from the huge matrices this pipeline produces.



Résumé

En utilisant des paquets R massivement parallèle des données de cytométrie en flux, l'objectif de ce rapport est d’exposer un pipeline nouvellement conçu qui permet aux biologistes n'ayant aucune connaissance préalable en programmation d'effectuer le nettoyage des données de cytométrie, l'échantillonnage et l'analyse de la surface cellulaire en utilisant des modèles d'apprentissage automatique. Ces outils offrent également diverses possibilités de visualisation ainsi que la génération de statistiques pour la validation des données. Le rapport présente également une exploitation réelle d'un ensemble de données et les étapes suivies pour extraire la signification biologique des énormes matrices produites par ce pipeline.